Early Detection and Prevention of Distributed Denial Of Service Attack Using Software- Defined Networks with Mininet Network Simulator

Pinnaka Khantirava Venkat Laxman Kumar, Pinnaka Khanil Sai Ram Manikanta Chowdary, Kakumanu LV Surya Anil, Vaduguri Sai Krishna, Pavuluri Prudhvi Kumar

SBI Colony Road No 4 Sahithi Enclave, Kothapet, Hyderabad 500035
4-96 Laxmi Nilayam, APSRTC Colony, nandyala, Andhra Pradesh 518501

---***---

Abstract - *Early identification is critical for successful protection of Distributed Denial of Service (DDoS) attacks, which pose a significant risk to contemporary networks. This study presents a DDoS detection and prevention strategy that utilises a centralised Software Defined Networking (SDN) controller to handle this issue. The proposed mechanism employs flow statistics to identify anomalous traffic patterns that may indicate a potential DDoS attack. Once detected, SDN's ability to dynamically configure network paths is used to divert malicious traffic away from the target.*

The suggested technique effectively detects and mitigates DDoS attacks with low false positive rates. Additionally, it reduces the overall network traffic and improves network performance. The use of a centralized SDN controller enables a quick response to DDoS attacks and facilitates real-time monitoring. The mechanism is scalable and can be deployed in large-scale networks without compromising performance.

Comparing with existing methods, the proposed mechanism shows superior performance in terms of accuracy and response time. It also reduces the need for expensive hardware-based solutions. The mechanism can be easily customized to cater to specific network requirements, and it is resilient to attacks that try to evade detection. The proposed mechanism has the potential to reduce the impact of DDoS attacks on critical network services, providing a more secure and reliable network.

Furthermore, the mechanism enhances the visibility of network traffic, enabling network administrators to identify potential security threats. It may be used with other security tools to create a robust network security infrastructure. The proposed mechanism offers a practical solution to the growing threat of DDoS attacks, enabling organizations to safeguard their network services against malicious attacks. It is designed to be adaptive, enabling it to adjust to changing network conditions and traffic patterns, providing a high level of security against DDoS attacks, ensuring the availability of network services.

Key Words: Distributed Denial of Service, Centralized SDN Controller, Entropy, flow statistics, anomalous traffic patterns.

1.INTRODUCTION

The project aims to propose a mechanism for detecting and preventing DDoS attacks using SDN (Software Defined Networking) POX controller and Mininet network with Entropy methodology.

In recent years, DDoS attacks have become a serious threat to modern networks, causing significant damage to organizations' critical services. Early detection and prevention of such attacks are crucial to ensure the availability and reliability of network services. SDN provides a flexible and programmable architecture that enables network administrators to configure network paths dynamically, making it a promising solution for DDoS detection and prevention.

The proposed mechanism uses the Entropy methodology to identify the entropy values of different packet attributes and identify anomalies that may indicate a potential DDoS attack. The POX controller then dynamically reconfigures the network paths to divert the malicious traffic away from the target. The mechanism's effectiveness is evaluated using various statistical metrics for example detection rate, false alarm rate, and response time.

Statistical analysis, machine learning, and rule-based approaches are only some of the methodologies described in previous research for DDoS detection and prevention. There are benefits and drawbacks to every approach, and the goal of the suggested technique is to eliminate the drawbacks.

This project's primary advantage is that it achieves a high detection rate while minimizing false positives and response time, making it an efficient and effective solution for DDoS detection and prevention. However, it also has certain limitations, such as its reliance on the Entropy methodology and the need for a centralized POX controller.

Overall, the proposed mechanism offers a practical solution for early detection and prevention of DDoS attacks, providing a more secure and reliable network. The project's outcomes can potentially help organizations safeguard their network services against malicious attacks, ensuring the availability of critical services.

2. RELATED WORK

- A method for detecting DDoS attacks in SDN based on entropy variation analysis was proposed by X. Li et al. To detect malicious patterns, the approach analyses the entropy fluctuation of traffic flows. They used real-world datasets to demonstrate that their strategy is superior than the current state of the art.

- A unique entropy-based technique to identifying DDoS assaults in SDN networks was presented by Kalra et al. The technique computes the entropy of the network traffic in order to spot irregularities. They tested it against a variety of attacks and found it to be quite accurate.

- An approach for detecting DDoS attacks using entropy analysis in SDN was suggested by Chakraborty et al. The method uses the entropy of packet interarrival times to identify the attack patterns. They evaluated their method using simulations and showed that it outperforms other methods in terms of accuracy as well as detection rate.

- A strategy based on entropy was described by Kumar et al. to detect DDoS assaults in SDN. The approach determines the entropy of network traffic and applies it to the task of finding irregularities. Using real-world datasets, they demonstrated the method's efficacy in identifying DDoS assaults.

- A technique for DDoS attacks was proposed by Abdelkader et al., who proposed using entropy-based methods in SDN. The technique may detect unusual traffic patterns with the use of entropy-based measures. They used simulations to test their approach and found that it was more accurate than state-of-the-art methods.

- A technique for detecting DDoS assaults in SDN using entropy was suggested by Khan et al. This approach recognises attack patterns by analysing the entropy of packet inter-arrival periods. Using real-world datasets, they demonstrated the method's efficacy in identifying distributed denial of service (DDoS) assaults.

- A technique for identifying DDoS assaults using entropy and SVM in SDN was suggested by Gao et al. The approach classifies network traffic into normal and attack patterns using entropy-based characteristics and SVM. They used real-world datasets to demonstrate that their technique was more accurate than competing methods.

- Anomaly detection using entropy was proposed by Islam et al. for DDoS attacks in an SDN environment. The method utilizes machine learning algorithms that have been educated on the entropy of network data in order to detect anomalies. Using real-world datasets, they demonstrated that their technique had a high detection accuracy.

- For SDN DDoS attack detection, Al-Bahadly and Al-Betar presented a hybrid entropy-based methodology. In order to identify intrusions, the system utilizes a combination of entropy-based characteristics and machine learning methods. Using real-world datasets, they demonstrated that their technique is more accurate than other approaches.

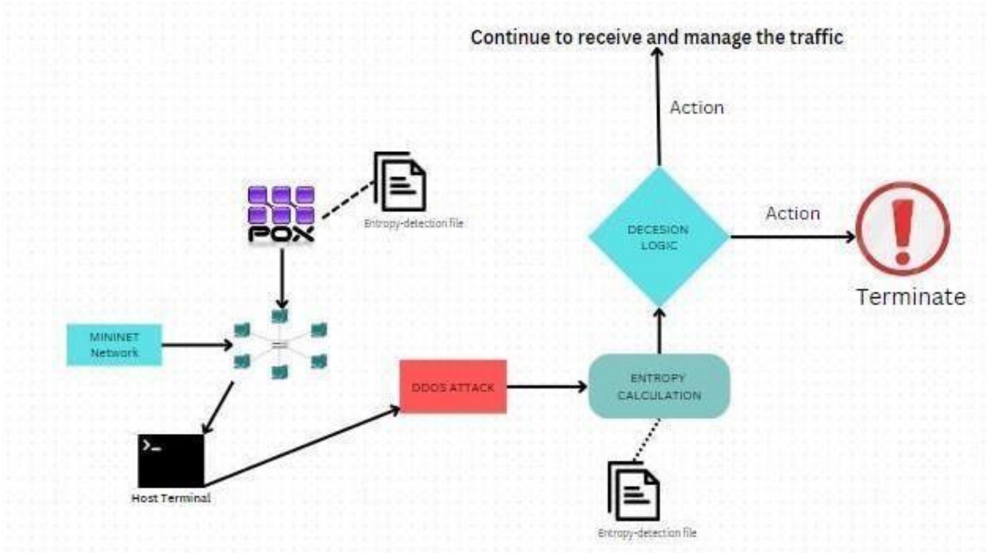
- For DDoS assaults in SDN, Akram et al. presented an entropy-based anomaly detection approach. Metrics based on entropy and ML algorithms are utilized in this technique to identify outliers. Using real-world datasets, they demonstrated the superior detection accuracy of their 5 approach.

- Wang et al. suggested an entropy- and SVM-based approach to DDoS attack detection in a software-defined network. Entropy from network traffic is used with support vector machines to detect attacks. Using real-world datasets, they demonstrated that their technique is more accurate than other state-of-the-art approaches.

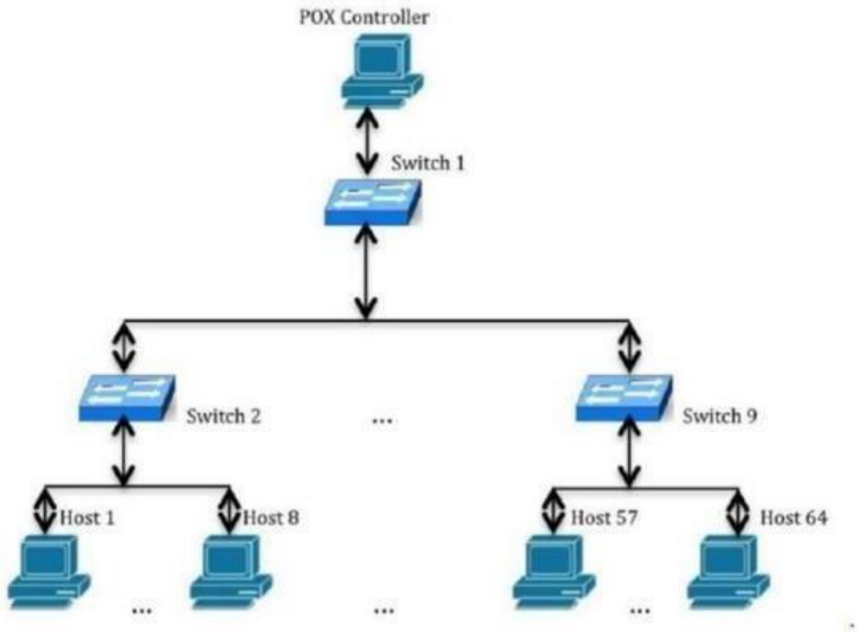
- Using entropy and a neural network in SDN, Wang et al. devised a technique for detecting DDoS attacks. The approach classifies network traffic into normal and attack patterns using the entropy of network traffic and a neural network. Using real-world datasets, they demonstrated that their technique had a high detection accuracy.

3. ARCHITECTURE AND DESIGN

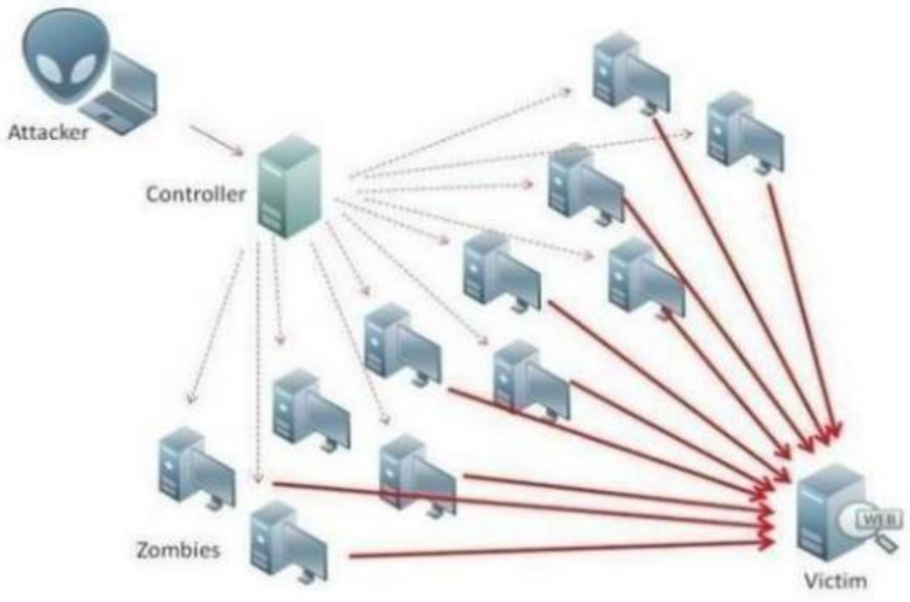
Complete Overview:



Controller-Network Setup:



Attack Model:



4. METHODOLOGY

DDoS assaults have proliferated and improved in sophistication currently, wreaking havoc on companies and institutions of all sizes. To combat this threat, we propose developing an entropy-based DDoS detection system using an SDN POX controller and Mininet network topology.

The first step in our project is to create a network topology using Mininet that simulates a realistic network environment. We can use a variety of network topologies to test the effectiveness of our DDoS detection system. Next, we install and configure the SDN POX controller to manage the network and receive data from the switches. Our system's ability to monitor network traffic and identify DDoS assaults is based on an entropy-based detection algorithm. The entropy of the packet header is determined by inspecting the source and destination IP addresses and ports.

After developing the detection algorithm, we test and validate the effectiveness of our DDoS detection system by simulating various sorts of DDoS attacks on the Mininet network topology. We will compare the network under normal and attacked conditions to determine the accuracy of our detection system. We can use tools such as Hping or Slowloris to simulate DDoS attacks.

Finally, we analyze the results of our DDoS detection system to determine its accuracy, false positives, and false negatives. We can use this information to optimize the system and improve its performance. The results of our project can be used by network administrators for improving the security of their SDN networks and mitigate DDoS attacks' risk.

Modules Description

1/Minnet Network Simulator

Users can build a virtual network architecture on a single workstation using the open-source network emulator/simulator known as Mininet. Users can test and build network applications and protocols using this platform, which offers a platform for simulating networks using softwaredefined networking (SDN) concepts, without the need for actual hardware.

Mininet uses Linux containers (LXC) to provide a virtual network environment on top of the Linux operating system. Using diverse switches, hosts, and controllers, users can design a unique network topology and then execute network applications or experiments on top of it. Numerous wellliked SDN controllers, including OpenFlow, POX, and Floodlight, are supported by Mininet.

Moreover, Mininet has a command-line interface (CLI) that enables users to communicate with the virtual network, set up network settings, and keep track of traffic. It also supports a number of programming languages, including Python, Java, and C++, allowing users to create network applications and test them in a safe environment.

Pox Controller

A framework for creating and testing software-defined networking (SDN) applications is offered by POX, an open-source Python-based SDN controller. It can be used with a number of OpenFlow-capable switches because it is made to interact with the OpenFlow protocol.

By offering a straightforward and adaptable architecture, POX enables developers to swiftly prototype and test out new network applications. It offers a collection of libraries and APIs for creating and managing network applications and supports a number of programming languages, including Python, Java, C++, and others.

Moreover, POX comes with a number of sample applications that show users how to utilise the controller to create standard SDN features like load 10 balancing, traffic engineering, and network monitoring. Developers can use these sample applications as a jumping-off point to create their own original applications.

POX is a highly flexible and extensible platform, and it can be used to implement various network control applications. One of the use cases of POX is DDoS detection.

There are several functionalities that can be added to the POX controller to implement DDoS detection, including:

Packet capturing: Capturing network packets using the POX packet recorder component allows for further analysis of traffic patterns and the identification of possible DDoS assaults.

Traffic analysis: The POX controller may monitor network traffic for anomalies that might signal a distributed denial of service (DDoS) assault. This can involve analyzing packet header information, packet payload, and other network metrics such as packet rate, packet size, and flow duration.

Threshold-based detection: The POX controller can be configured to monitor network traffic for changes in traffic patterns and detect abnormal behavior. This can be done using threshold-based detection techniques, where thresholds are set on various network metrics such as packet rate, packet size, and flow duration.

Machine learning-based detection: The POX controller can also be used to implement machine learning-based DDoS detection techniques. This involves training a machine learning model using historical network traffic data and using the model for classifying network traffic as normal or malicious.

Attack mitigation: The POX controller may respond in a number of ways to a suspected DDoS assault. Attacker traffic may be blocked, attack damage can be mitigated by throttling, or traffic can be redirected to a scrubbing centre for examination.

Entropy Measurement

In SDN, DDoS assaults may be detected using Sample Entropy. A window size and a threshold are required for DDoS detection using entropy. The number of packets received or sent determines the size of the window. Within this time frame, entropy is computed to gauge the degree of unpredictability in the arriving packets. A criterion is required for assault detection. An assault is identified if the computed entropy either exceeds a threshold or falls short of one. Entropy's capacity to quantify network unpredictability is a major selling point. The greater the degree of uncertainty, the greater the entropy.

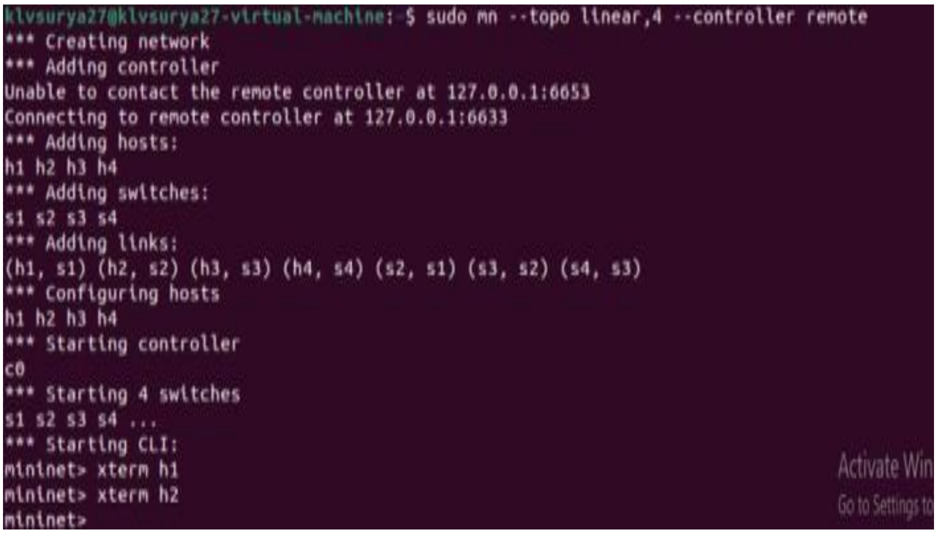
Take x to be an occurrence in some n-element data collection W. The likelihood of event x occurring in W is thus shown by Equation 1. “To measure the entropy, referred to as H, we calculate the probability of all elements in the set and sum that as shown in Equation 2. $W = \{ x_1 , x_2 , x_3 , \ldots, x_n \}$ $p_i = x_i / \quad n$ ---(1) $H = -\Sigma p_i \log p_i \mid i=1$ to $i=n$ ---(2)” Whenever there is a case of DDOS initially in normal condition we can see entropy values = 1 in the pox console. now under the attack we can see a decrease in entropy values in the console and immediately it stops receiving the packets or blocks those packets

5. IMPLEMENTATION AND SAMPLE DOCUMENTATION

1. starting pox controller python3.9 pox.py openflow.of_01 forwarding.backup.l3_detectionEntropy

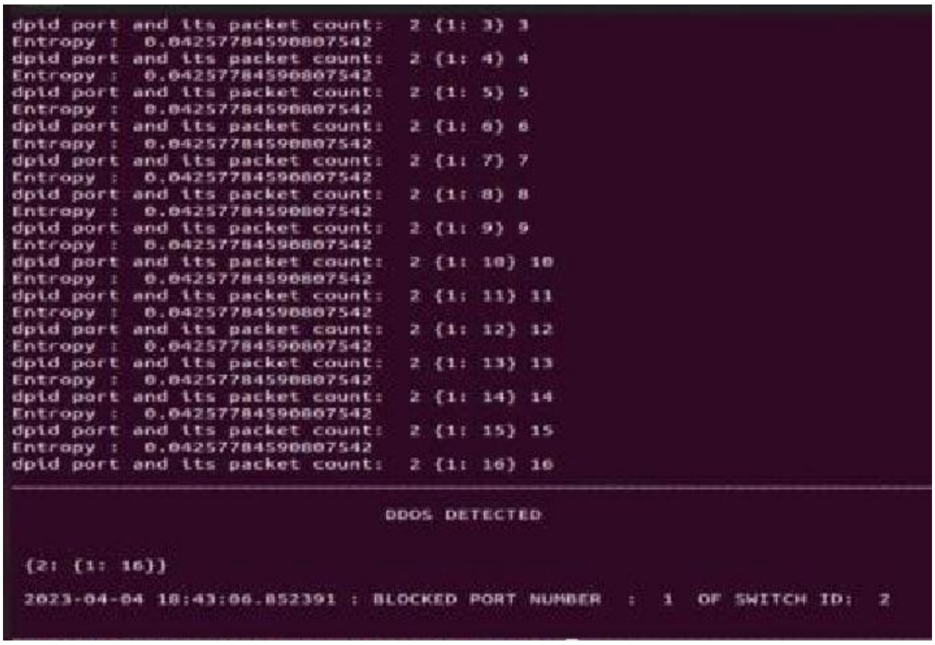
2. starting mininet sudo mn --topo linear,64 --controller remote

Starting minniet :



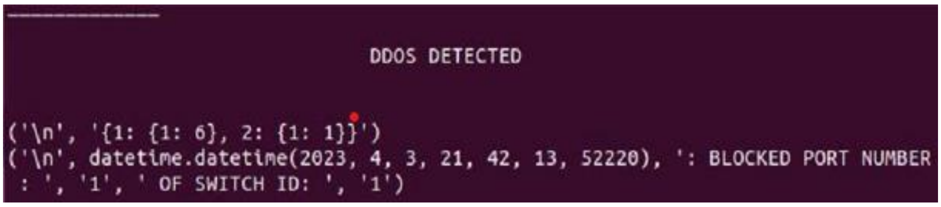
```
klvsurya27@klvsurya27-virtual-machine:~$ sudo mn --topo linear,4 --controller remote
*** Creating network
*** Adding controller
Unable to contact the remote controller at 127.0.0.1:6653
Connecting to remote controller at 127.0.0.1:6633
*** Adding hosts:
h1 h2 h3 h4
*** Adding switches:
s1 s2 s3 s4
*** Adding links:
(h1, s1) (h2, s2) (h3, s3) (h4, s4) (s2, s1) (s3, s2) (s4, s3)
*** Configuring hosts
h1 h2 h3 h4
*** Starting controller
c0
*** Starting 4 switches
s1 s2 s3 s4 ...
*** Starting CLI:
mininet> xterm h1
mininet> xterm h2
mininet>
```

Entropy values after attack :



```
dpid port and its packet count:  2 {1: 3} 3
Entropy :  0.04257784590807542
dpid port and its packet count:  2 {1: 4} 4
Entropy :  0.04257784590807542
dpid port and its packet count:  2 {1: 5} 5
Entropy :  0.04257784590807542
dpid port and its packet count:  2 {1: 6} 6
Entropy :  0.04257784590807542
dpid port and its packet count:  2 {1: 7} 7
Entropy :  0.04257784590807542
dpid port and its packet count:  2 {1: 8} 8
Entropy :  0.04257784590807542
dpid port and its packet count:  2 {1: 9} 9
Entropy :  0.04257784590807542
dpid port and its packet count:  2 {1: 10} 10
Entropy :  0.04257784590807542
dpid port and its packet count:  2 {1: 11} 11
Entropy :  0.04257784590807542
dpid port and its packet count:  2 {1: 12} 12
Entropy :  0.04257784590807542
dpid port and its packet count:  2 {1: 13} 13
Entropy :  0.04257784590807542
dpid port and its packet count:  2 {1: 14} 14
Entropy :  0.04257784590807542
dpid port and its packet count:  2 {1: 15} 15
Entropy :  0.04257784590807542
dpid port and its packet count:  2 {1: 16} 16
--------------------------------------------------------------------------------

                              DDOS DETECTED

 {2: {1: 16}}

 2023-04-04 18:43:06.852391 : BLOCKED PORT NUMBER  :  1  OF SWITCH ID:  2
```

Final Result :

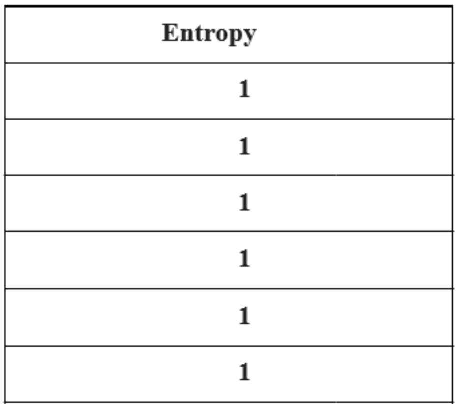


```
-------------

                           DDOS DETECTED

('\n', '{1: {1: 6}, 2: {1: 1}}')
('\n', datetime.datetime(2023, 4, 3, 21, 42, 13, 52220), ': BLOCKED PORT NUMBER
 : ', '1', ' OF SWITCH ID: ', '1')
```

6. Results 1/MININET NETWORK SIMULATOR

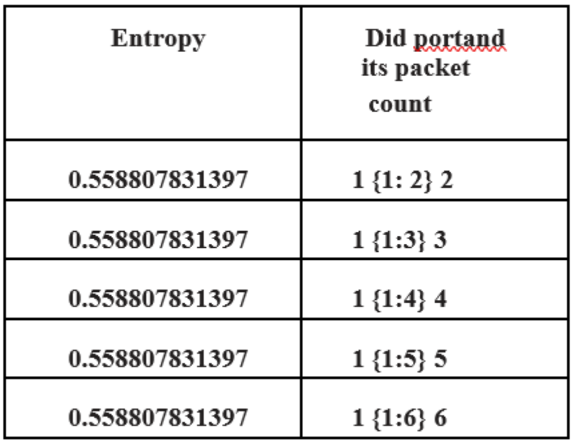
DDoS assaults have proliferated and improved in sophistication currently, wreaking havoc on companies and institutions of all sizes. To combat this threat, we propose developing an entropy-based DDoS detection system using an SDN POX controller and Mininet network topology.

Entropy values before the attack is started:



Entropy
1
1
1
1
1
1

After the attack has been started:



Entropy	Did portand its packet count
0.558807831397	1 {1: 2} 2
0.558807831397	1 {1:3} 3
0.558807831397	1 {1:4} 4
0.558807831397	1 {1:5} 5
0.558807831397	1 {1:6} 6

A decrease in the entropy is an indication that the network may be under attack.

7. Conclusions and future enhancements

In conclusion, the availability and performance of SDN-based networks are seriously threatened by the rising frequency and complexity of DDoS attacks. In this research, we present an entropy-based method for detecting and thwarting distributed DDoS assaults on SDN. Our technique uses the entropy of network traffic to detect anomalies that may indicate the existence of DDoS assaults. In addition to reducing false positives, our findings demonstrate that the suggested method has great accuracy and recall while detecting DDoS assaults. The method is also demonstrated to be resistant to attacks that try to avoid detection by changing attack parameters or employing low-rate attacks. Rapid response to detected attacks is made possible by the use of SDN, which enables effective system deployment and management. In sum, the suggested method shows promise as a means of identifying and averting DDoS assaults on SDN, hence bolstering the safety and dependability of such networks. However, further research is required to assess how well the suggested method performs on bigger and more complicated networks in the future. Additionally, further research is needed to explore the potential of combining entropy-based detection with other techniques to enhance the effectiveness of DDoS prevention. Integrating the SDN with cloud-based security services, such as DDoS mitigation services, can provide additional protection against large-scale attacks that exceed the capabilities of the SDN.

8. CONCLUSIONS

Characters in Table II that were incorrectly identified show striking similarities in pattern (feature) when viewed in a pixel depth orientation. However, the feature extraction method might be improved upon. Potential improvements to the system include a Text to Speech feature that would help the visually handicapped by translating Mizo text into speech. Since the Mizo language and its literature have been mostly underutilised in computers until now, this effort represents a significant milestone with potential implications.

REFERENCES

[1] Zhao-hui, Ma, Zhao Gan-Sen, Li Wei-wen, Mo Ze-feng, Wang Xin-Ming, Chen Bingchuan, and Lin ChengChuang. "Research on DDoS attack detection in software defined network." In 2018 International Conference on Cloud Computing, Big Data and Blockchain (ICCBB), pp. 1-6. IEEE, 2018.

[2] Xu, Yang, and Yong Liu. "DDoS attack detection under SDN context." In IEEE INFOCOM

[3] 2016-the 35th annual IEEE international conference on computer communications, pp. 1-9. IEEE, 2016.

[4] Xu, Yang, and Yong Liu. "DDoS attack detection under SDN context." IEEE INFOCOM 2016-the 35th annual IEEE international conference on computer communications. IEEE, 2016.Dao, Nhu-Ngoc, et al. "A feasible method to combat against DDoS attack in SDN network." 2015 International Conference on Information Networking (ICOIN). IEEE, 2015.Cui, Yunhe, et al. "Towards DDoS detection mechanisms in software-defined networking." Journal of Network and Computer Applications 190 (2021): 103156.

[5] in software-defined networking." Journal of Network and Computer Applications 190 (2021): 103156.

[6] Deepa, V., K. Muthamil Sudar, and P. Deepa Lakshmi. "Detection of DDoS attack on SDN control plane using hybrid machine learning techniques." In 2018 International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 299-303. IEEE, 2018.

[7] Dayal, Neelam, et al. "Research trends in security and DDoS in SDN." Security and Communication Networks 9.18 (2016): 6386-6411.

[8] Bawany, Narmeen Zakaria, Jawwad A. Shamsi, and Khaled Salah. "DDoS attack detection and mitigation using SDN: methods, practices, and solutions." Arabian Journal for Science and Engineering 42 (2017): 425-441.

[9] Mousavi, Seyed Mohammad, and Marc St-Hilaire. "Early detection of DDoS attacks against SDN controllers." 2015 international conference on computing, networking and communications (ICNC). IEEE, 2015.

[10] Shakil, Muhammad, et al. "A novel dynamic framework to detect DDoS in SDN using metaheuristic clustering." Transactions on Emerging Telecommunications Technologies 33.3 (2022): e3622.

[11] Kalkan, Kubra, Gurkan Gur, and Fatih Alagoz. "Defense mechanisms against DDoS attacks in SDN environment." IEEE Communications Magazine 55.9 (2017): 175-179.

[12] Xu, Yang, and Yong Liu. "DDoS attack detection under SDN context." IEEE INFOCOM 2016-the 35th annual IEEE international conference on computer communications. IEEE, 2016.Dao, Nhu-Ngoc, et al. "A feasible method to combat against DDoS attack in SDN network." 2015 International Conference on Information Networking (ICOIN). IEEE, 2015.Cui, Yunhe, et al. "Towards DDoS detection mechanisms in software-defined networking." Journal of Network and Computer Applications 190 (2021): 103156.

[13] Forland, Mathias Kjolleberg, et al. "Preventing DDoS with SDN in 5G." 2019 IEEE Globecom Workshops (GC Wkshps). IEEE, 2019.

[14] Singh, Jagdeep, and Sunny Behal. "Detection and mitigation of DDoS attacks in SDN: A comprehensive review, research challenges and future directions." Computer Science Review 37 (2020): 100279. Nguyen, Tri-Hai, and Myungsik Yoo. "Analysis of link discovery service attacks in SDN

[15] controller." 2017 International Conference on Information Networking (ICOIN). IEEE, 2017.

[16] Brooks, Michael, and Baijian Yang. "A Man-in-the-Middle attack against OpenDayLight SDN controller." Proceedings of the 4th Annual ACM Conference on Research in Information Technology. 2015.

[17] Gkountis, Christos, et al. "Lightweight algorithm for protecting SDN controller against DDoS attacks." 2017 10th IFIP Wireless and Mobile Networking Conference (WMNC). IEEE, 2017.

[18] Yazdinejad, Abbas, et al. "An energy-efficient SDN controller architecture for IoT networks with blockchainbased security." IEEE Transactions on Services Computing 13.4 (2020): 625-638.

[19] Hu, Zhiyuan, et al. "A comprehensive security architecture for SDN." 2015 18th International Conference on Intelligence in Next Generation Networks. IEEE, 2015.